WEST Search History

Get Items | Restore | Clear | Cancel

DATE: Tuesday, November 30, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR	
☐	L8	L3 with reboot$4	19
☐	L7	l3 same (access$4 near3 (quick$4 or fast$4 or speed or rate))	9
☐	L6	L4 same reload$4	4
☐	L5	L4 same (load$4 or reload$4)	63
☐	L4	L3 same (boot$4 or reboot$4)	166
☐	L3	((copies or copy) near5 (os or (operating adj system)))	2602
☐	L2	L1 same (boot$4 or reboot$4)	2
☐	L1	(sav$4 near5 (copies or copy) near5 (os or (operating adj system)))	22

END OF SEARCH HISTORY

L2: Entry 1 of 2 File: USPT Oct 5, 1999

DOCUMENT-IDENTIFIER: US 5961642 A
TITLE: Generic kernel modification for the dynamic configuration of operating systems in a multi-processor system

Detailed Description Text (20):
After the changes have been applied, Block 40 represents the Cross-Package-Add Tool 22 shutting down each of the processors 12 of the MPP computer system 10, (optionally) saving a copy of the previous operating system 18 to another location on the data storage devices 16 associated with the processors 12, and booting the modified generic kernel of the new operating system 18 from the root locations 20 of the data storage devices 16 into their respective processors 12 of the MPP computer system 10.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

End of Result Set

L2: Entry 2 of 2 File: USPT Jan 20, 1998

DOCUMENT-IDENTIFIER: US 5710930 A
TITLE: Apparatus and a method for allowing an operating system of a computer system to persist across a power off and on cycle

Detailed Description Text (26):
At step 53, the shutdown program is relocated to or made sure to be in the low address memory (i.e., lower than the one megabyte address range constraint) of system memory 23. The process then goes to step 54. At step 54, the current system state of computer system 20 is stored in shutdown state buffer 43 (see FIG. 2). As described above, the current system state of computer system 20 typically includes data stored in frame buffer 24, data held in various registers 35 of microprocessor 22, etc. The shutdown program also saves a copy of the operating system loader program in the shutdown state buffer. In addition, the shutdown program also retains shutdown state buffer 43. The process then moves to step 55. At step 55, the loader program (i.e., initialization code) stored in boot sector 46 of mass storage device 26 is replaced with a new loader program. The new loader program is created by the shutdown program of the present invention that contains the address to the shutdown program's "restart" entry point. This means that the new loader program contains the "restart" entry address of the shutdown program and will branch to the shutdown program at system power up. The shutdown program, when executed, will access shutdown state buffer 43 (FIG. 2) for the data representing the state of computer system 20 before power-off. The shutdown program then sends the system state data to frame buffer 24, registers 35, etc. to restore computer system 20 to the state where it was powered off. The process then moves to step 57.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

L6: Entry 1 of 4 File: USPT May 8, 2001

DOCUMENT-IDENTIFIER: US 6230286 B1
TITLE: Computer system failure reporting mechanism

Brief Summary Text (5):
Much of the world's day-to-day business relies on sophisticated computer systems used in such fields as finance, securities exchange, telecommunications, medicine, and retailing. Reliability and maintainability are vital concerns in such applications, since any down time may result not only in tremendous inconvenience but also significant monetary loss. Vendors of such computer systems have therefore developed maintenance and diagnostic facilities as part of the computer system and have established customer assistance centers to provide customer support. When a system failure occurs, diagnostic software tries to determine the cause of the failure and sends analysis results to the customer assistance center. In the prior art, reporting mechanisms of the type described have typically been required to run under the computer operating system, such that if a failure occurs, the system must be able to be successfully rebooted before failure reporting can be performed. To ensure that the system can be successfully rebooted, redundant boot paths have been provided. That is, multiple independent copies of the operating system have been stored on different storage devices independently accessible by the computer system such that if the operating system cannot be reloaded from one of the storage devices, then it may hopefully be reloaded from another of the storage devices. Despite such precautions, failures do occur that prevent the system from being successfully rebooted. In order to minimize down time, a mechanism is needed for reporting such failures. Such a mechanism would allow a "dead" system to "call home"; that is, allow a system that cannot be successfully rebooted to send a report to a remote service center.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

L6: Entry 2 of 4 File: USPT Aug 10, 1999

DOCUMENT-IDENTIFIER: US 5935242 A
TITLE: Method and apparatus for initializing a device

Brief Summary Text (11):
Embodiments of the present invention provide a system for controlling the rebooting of a device such as a computer. When a warm reboot is triggered, the invention provides a mechanism for determining whether a valid copy of the operating system and related files are already stored in a memory device in the computer. The system avoids reloading the entire operating system if a valid copy of the operating system is already stored in the computer. The system does not require the additional cost and complexity associated with systems using Flash ROMs. This is particularly important in cost-sensitive devices such as network-booted computers and other network-booted devices.

Previous Doc Next Doc Go to Doc#

First Hit Previous Doc Next Doc Go to Doc#

End of Result Set

Generate Collection Print

L6: Entry 4 of 4 File: EPAB Apr 10, 2002

DOCUMENT-IDENTIFIER: EP 1195679 A1
TITLE: Performing operating system recovery from external back-up media in a headless computer entity

Abstract Text (1):
CHG DATE=20020503 STATUS=O> A computer entity, particularly but not exclusively a headless computer entity, has operating systems stored on a non-volatile data storage device e.g. a hard disk drive, and has a back-up data storage device. Operating system back-up's are taken from an uncorrupted copy of an operating system stored in a separate partition on the data storage device to the primary operating system which is actually used to run the device, thereby ensuring that if the primary operating system of the computer entity becomes corrupted either gradually or catastrophically, the back-up copy which is stored on a back-up media is not effected. Under failure conditions of the computer entity, a pristine copy of the operating system can be reloaded from the back-up tape data storage media and the computer entity rebooted from the pristine operating system back-up copy.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

L8: Entry 3 of 19 File: USPT Feb 10, 2004

DOCUMENT-IDENTIFIER: US 6691244 B1
TITLE: System and method for comprehensive availability management in a high-availability computer system

Detailed Description Text (4):
In one embodiment, each node 102, 104 and 106 contains a copy of the operating system 112 used within the cluster 100. A copy of the operating system 112 is stored in executable memory, and may be rebooted from disk storage (not shown) or from a computer network connected to the cluster 100. The operating system 112 may also be stored in nonvolatile random access memory (NVRAM) or flash memory. Individual nodes 102, 104 and 106 can each be rebooted with no effect on the other nodes.

Previous Doc Next Doc Go to Doc#

L8: Entry 8 of 19 File: USPT Nov 13, 2001

DOCUMENT-IDENTIFIER: US 6317826 B1
TITLE: Booting a computer system from a network

Detailed Description Text (18):
Switchset.bat, listed below, determines which of two network drivers is required based on a program, "Checknet", and sets up the appropriate config.sys and protocol.ini files for the required drivers. Switchset also copies autoexec.nxt to autoexec.bat to set up the process to be executed on the next boot, then runs "localboot" which restores the interrupt vector environment from memory and reboots from the modified copy of the operating system in the RAM drive 24.